AMENDMENTS

In the Claims

1. (Previously Presented) A computer implemented method for scheduling comprising the steps of:

scheduling a resource among a plurality of elements by:

detecting expiration of a period-of-use of said resource, said resource allocated to an active one of said plurality of elements for said period-of-use;

updating a measure-of-use of said resource for said resource for said active one of said plurality of elements responsive to said period-of-use and a measure-of-use adjustment; and

assigning one of said plurality of elements to use said resource for a second period-of-use responsive to said measure-of-use and an element-specific selection adjustment for each element in said plurality of elements, wherein said element-specific selection adjustment for said each element in said plurality of elements is borrowed virtual time.

2. (Original) The method of claim 1 wherein said period-of-use is a scheduled period-of-use.

3. (Original) The method of claims 1 wherein said plurality of elements is a plurality of threads-of-execution and said resource is time available to a central processor unit (CPU) to execute said plurality of threads-of-execution.

4. (Previously Presented) The method of claim 3 wherein the step of updating said measure-of-use further includes updating a virtual time for said active one of said plurality of threads-of-execution responsive to said period-of-use; and wherein the step of assigning one of said plurality of elements further includes determining an effective virtual time responsive to said virtual time and said element-specific selection adjustments.

5. (Original) The method of claim 4 further including specifying said borrowed virtual time by one of said plurality of threads-of-execution.

6. (Original) The method of claim 3 further including steps of:
adding a new thread to said plurality of threads-of-execution by a parent thread; and
initializing said virtual time for said new thread using said virtual time of said parent thread.

7. (Original) The method of claim 3 wherein said plurality of threads-of-execution includes a set of ready threads and a set of blocked threads.

8. (Original) The method of claim 7 wherein said method further includes adjusting each of said set of blocked threads by an adjustment value.

9. (Original) The method of claim 7 wherein said method further includes updating a system reference-use of said resource.

10. (Original) The method of claim 9 wherein said method further includes steps of:
determining that one of said set of blocked threads has become ready; and
updating, responsive to the step of determining, a virtual time for said one of said set of blocked threads or to said system reference-use as adjusted by a lag limit.

11. (Original) The method of claim 9 wherein said method further includes steps of:
(a) determining that one of said set of blocked threads had become blocked;
(b) saving said system reference-use and a current real-time value associated with said one of said set of blocked threads;
(c) determining that said one of said set of blocked threads has become ready; and
(d) updating a virtual time for said one of said set of blocked threads responsive to step (c) and further responsive to said saved system reference-use, said saved current real-time, and said system reference-use.

12. (Original) The method of claim 9 whereby said system reference-use is updated to converge towards a virtual time average over said set of ready threads.

13. (Original) The method of claim 12 wherein the step of updating said system reference-use is accomplished substantially in accordance with:

```
reference_use=max(reference_use,
                  min(reference_use+R+RCost, EVT));
```

where reference_use is said system reference-use, R is a convergence rate, RCost is a resource usage, and EVT is an effective virtual time, and said resource usage is a function of said period-of-use and said measure-of-use adjustment assigned to said active one of said plurality of threads-of-execution.

14. (Original) The method of claim 12 wherein the step of updating said system reference-use is accomplished substantially in accordance with:

```
reference_use += max(-MaxChange,
                     min(MaxChange, MeanAVT-reference_use));
```

where reference_use is said system reference-use, MaxChange is responsive to a resource usage, and MeanAVT is an average AVT over a set of said plurality of elements, and said resource usage is a function of said period-of-use and said measure-of-use adjustment assigned to said active one of said plurality of threads-of-execution.

15. (Original) The method of claim 9 further including steps of:
adding a new thread to said plurality of threads-of-execution; and
initializing said virtual time for said new thread using said system reference-use.

16. (Previously Presented) The method of claim 1 wherein the said plurality of elements is a plurality of queues and said resource is a bandwidth of an output port of a data switch.

17. (Previously Presented) The method of claim 16 wherein the step of updating said measure-of-use further includes updating a virtual time for said active one of said plurality of

queues responsive to said period-of-use; and wherein the step of assigning one of said plurality of elements further includes determining an effective virtual time responsive to said virtual time and said element-specific selection adjustment.

18. (Original) The method of claim 16 wherein said period-of-use is a transmission time period required to transfer one or more data packets from one of said plurality of queues to said output port.

19. (Original) The method of claim 16 wherein said plurality of queues includes a set of non-empty queues and a set of empty queues.

20. (Original) The method of claim 19 wherein said method further includes updating a system reference-use of said resource.

21. (Original) The method of claim 20 wherein said method further includes steps of:

(a) determining that one of said set of non-empty queues has become empty;

(b) saving said system reference-use and a current real-time value associated with said now-empty queue;

(c) determining that said now-empty queue has become non-empty; and

(d) updating a virtual time for said now-non-empty queue responsive to step (c) and further responsive to said saved system reference-use, said saved current real-time, and said system reference-use.

22. (Original) The method of claim 20 whereby said system reference-use is updated to converge towards a virtual time average over said set of non-empty queues.

23. (Original) The method of claim 22 wherein the step of updating said system reference-use is accomplished substantially in accordance with:

```
reference_use = max(reference_use,
                    min(reference_use+R+RCost, EVT));
```

where reference_use is said system reference-use, R is a convergence rate, RCost is a resource usage, and EVT is said effective virtual time, and said resource usage is

a function of said period-of-use and a weight assigned to said active one of said plurality of queues.

24. (Original) The method of claim 22 wherein the step of updating said system reference-use is accomplished substantially in accordance with:

reference_use += max(-MaxChange,
min(MaxChange, MeanAVT-reference_use));

where reference_use is said system reference-use, MaxChange is responsive to a resource usage, and MeanAVT is an average AVT over at set of said plurality of elements, and said resource usage is a function of said period-of-use and a weight assigned to said active one of said plurality of queues.

25. (Original) The method of claim 20 further including steps of:

adding a new queue to said plurality of queues; and

initializing said virtual time for said new queue using said system reference-use.

26. (Original) The method of claim 20 wherein the step of updating alters said system reference-use using an adjustment value and said method further includes adjusting each of said set of empty queues by said adjustment value when said system reference-use is updated.

27. (Original) The method of claim 26 whereby said system reference-use is updated to converge towards a virtual time average over said set of non-empty queues.

28. (Previously Presented) The method of claim 27 wherein the step of updating said system reference-use is accomplished substantially in accordance with:

reference_use = max(reference_use,
min(reference_use+R+RCost, EVT));

where reference_use is said system reference-use, R is a convergence rate, RCost is a resource usage, and EVT is said effective virtual time, and said resource usage is a function of said period-of-use and a weight assigned to said active one of said plurality of queues.

29. (Original) The method of claim 27 wherein the step of updating said system reference-use is accomplished substantially in accordance with:

reference_use += max(-MaxChange,
min(MaxChange, MeanAVT-reference_use));

where reference_use is said system reference-use, MaxChange is responsive to a resource usage, and MeanAVT is an average AVT over at set of said plurality of elements, and said resource usage is a function of said period-of-use and a weight assigned to said active one of said plurality of queues.

30. (Currently Amended) ~~A computer-implemented scheduling apparatus comprising:~~

~~an apparatus for scheduling a resource among a plurality of elements, said apparatus including:~~ **A scheduling apparatus for scheduling a computer resource among a plurality of elements, comprising:**

a detection mechanism configured to detect expiration of a period-of-use of said resource, said resource allocated to an active one of said plurality of elements for said period-of-use;

an update mechanism configured to update a measure-of-use of said resource for said active one of said plurality of elements responsive to said period-of-use and a measure-of-use adjustment; and

an assignment mechanism configured to assign one of said plurality of elements to use said resource for a second period-of-use responsive to said measure-of-use and an element-specific selection adjustment for each element in said plurality of elements, wherein said element-specific selection adjustment for said each element in said plurality of elements is borrowed virtual time.

31. (Original) The apparatus of claim 30 wherein said period-of-use is a scheduled period-of-use.

32. (Original) The apparatus of claim 30 further including a central processing unit (CPU) and a memory coupled to said CPU, wherein said plurality of elements is a plurality of

threads-of-execution and said resource is time available to said CPU to execute said plurality of threads-of-execution.

33. (Previously Presented) The apparatus of claim 32 wherein the update mechanism further includes an update thread virtual time mechanism configured to update a virtual time for said active one of said plurality of threads-of-execution responsive to said period-of-use; and wherein the assignment mechanism further includes an effective virtual time determination mechanism configured to determine an effective virtual time responsive to said virtual time and said borrowed virtual time.

34. (Original) The apparatus of claim 33 further including a borrowed time specification mechanism configured to specify said borrowed virtual time by one of said plurality of threads-of-execution.

35. (Original) The apparatus of claim 32 further including:

a thread creation mechanism configured to add a new thread to said plurality of threads-of-execution by a parent thread; and

a virtual time initialization mechanism configured to initialize said virtual time for said new thread using said virtual time of said parent thread.

36. (Original) The apparatus of claim 32 wherein said plurality of threads-of-execution includes a set of ready threads and a set of blocked threads.

37. (Original) The apparatus of claim 36 further including a blocked thread update mechanism configured to adjust each of said set of blocked threads by an adjustment value.

38. (Original) The apparatus of claim 36 further including a reference-use update mechanism configured to update a system reference-use of said resource.

39. (Original) The apparatus of claim 38 further including:

a blocked determination mechanism configured to determine that one of said set of blocked threads had become blocked;

a capture mechanism, responsive to the blocked determination mechanism, configured to save said system reference-us and a current real-time value associated with said one of said set of blocked threads.

40. (Original) The apparatus of claim 38 wherein the reference-use update mechanism alters said system reference-use using an adjustment value and said apparatus further includes a virtual time update mechanism configured to adjust each of said set of blocked threads by said adjustment value substantially when said system reference-use is updated.

41. (Original) The apparatus of claim 38 whereby the reference-use update mechanism updates said system reference-use so that said system reference-use converges to virtual time average over said set of ready threads.

42. (Original) The apparatus of claim 41 wherein the reference-use update mechanism substantially implements:

```
reference_use = max(reference_use,
                    min(reference_use+R+RCost, EVT));
```

where reference_use is said system reference-use, R is a convergence rate, RCost is a resource usage, and EVT is said effective virtual time, and said resource usage is a function of said period-of-use and a weight assigned to said active one of said plurality of threads-of-execution.

43. (Original) The apparatus of claim 41 wherein the reference-use update mechanism substantially implements:

```
reference_use += max(-MaxChange,
                     min(MaxChange, MeanAVT-reference_use));
```

where reference_use is said system reference-use, MaxChange is responsive to a resource usage, and MeanAVT is an average AVT over at set of said plurality of elements, and said resource usage is a function of said period-of-use and a weight assigned to said active one of said plurality of threads-of-execution.

44. (Original) The apparatus of claim 38 further including:

a thread creation mechanism configured to add a new thread to said plurality of threads-of-execution; and

a virtual time initialization mechanism configured to initialize said new thread using said system reference-use.

45. (Previously Presented) The apparatus of claim 30 wherein said plurality of elements is a plurality of queues and said resource is a bandwidth of an output port of a data switch.

46. (Original) The apparatus of claim 45 wherein the update mechanism further includes an update queue virtual time mechanism configured to update a virtual time for said active one of said plurality of queues responsive to said period-of-use; and wherein the assignment mechanism further includes an effective virtual time determination mechanism configured to determine an effective virtual time responsive to said virtual time and said borrowed virtual time.

47. (Original) The apparatus of claim 45 wherein said period-of-use is a transmission time period required to transfer one or more data packets from one of said plurality of queues to said output port.

48. (Original) The apparatus of claim 45 wherein said plurality of queues includes a set of non-empty queues and a set of empty queues.

49. (Original) The apparatus of claim 48 wherein and said apparatus further includes a reference-use update mechanism configured to update a system reference-use of said resource.

50. (Original) The apparatus of claim 49 further including:

a blocked determination mechanism configured to determine that one of said set of non-empty queues has become empty;

a capture mechanism configured to save said system reference-use and a current real-time value associated with non-empty queue;

a ready determination mechanism configured to determine said now-empty queue has become non-empty; and

a virtual time update mechanism configured to update a virtual time for said now-non-empty queue responsive to the ready determination mechanism and further responsive to said saved system reference-use, said saved current real-time, and said system reference-use.

51. (Original) The apparatus of claim 49 wherein the reference-use update mechanism alters said system reference-use using an adjustment value and said apparatus further includes a virtual time update mechanism configured to adjust each of said set of empty queues by said adjustment value when said system reference-use is updated.

52. (Original) The apparatus of claim 49 wherein the reference-use update mechanism substantially implements:

```
reference_use = max(reference_use,
                    min(reference_use+R+RCost, EVT));
```

where reference_use is said system reference-use, R is a convergence rate, RCost is a resource usage, and EVT is said effective virtual time, and said resource usage is a function of said period-of-use and a weight assigned to said active one of said plurality of queues.

53. (Original) The apparatus of claim 49 wherein the reference-use update mechanism substantially implements:

```
reference_use += max(-MaxChange,
                     min(MaxChange, MeanAVT-reference_use));
```

where reference_use is said system reference-use, MaxChange is responsive to a resource usage, and MeanAVT is an average AVT over at set of said plurality of elements, and said resource usage is a function of said period-of-use and a weight assigned to said active one of said plurality of queues.

54. (Original) The apparatus of claim 49 further including:

a thread creation mechanism configured to add a new queue to said plurality of queues; and

a virtual time initialization mechanism configured to initialize said virtual time for said new queue using said system reference-use.

55. (Previously Presented) A computer program product including:

a computer usable storage medium having computer readable code embodied therein for scheduling a resource among a plurality of elements, said computer readable code including:

computer readable program code configured to cause said computer to effect a detection mechanism configured to detect expiration of a period-of-use of said resource, said resource allocated to an active one of said plurality of elements for said period-of-use;

computer readable program code configured to cause said computer to effect an update mechanism configured to update a measure-of-use of said resource for said active one of said plurality of elements responsive to said period-of-use and a measure-of-use adjustment; and

computer readable program code configured to cause said computer to effect an assignment mechanism configured to assign one of said plurality of elements to use said resource for a second period-of-use responsive to said measure-of-use and an element-specific selection adjustment for each element in said plurality of elements, wherein said element-specific selection adjustment for said each element in said plurality of elements is borrowed virtual time.

56. (Original) The computer program product of claim 55 wherein said period-of-use is a scheduled period-of-use.

57. (Original) The computer program product of claim 55 wherein said plurality of elements is a plurality of threads-of-execution and said resource is time available to said CPU to execute said plurality of threads-of-execution.